Zenith Week 4

LTI Systems

$$u \longrightarrow \boxed{\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}} \longrightarrow y$$

Let's figure out how such systems behave.

Start by ignoring the input term:

$$\dot{x} = Ax$$
$$x(t_0) = x_0$$

What will the solution to this system be?

Considering all values to be scalar we get:

$$\dot{x} = ax, x(t_0) = x_0 \Rightarrow x(t) = e^{a(t-t_0)}x_0$$

We know this because:

$$x(t_0) = e^{a(t_0-t_0)}x_0 = e^0 x_0 = x_0$$ ☑ initial conditions

$$\frac{d}{dt}x(t) = ae^{a(t-t_0)}x_0 = ax$$ ☑ dynamics

For higher-order systems we get a matrix version of this:

$$\dot{x} = Ax, x(t_0) = x_0 \Rightarrow x(t) = e^{A(t-t_0)}x_0$$

Matrix exponential term

The definition of this matrix exponential term is just like scalar exponentials:

$$e^{at} = \sum_{k=0}^{\infty} \frac{a^k t^k}{k!}, e^{At} = \sum_{k=0}^{\infty} \frac{A^k t^k}{k!}$$

Derivative:

$$\frac{d}{dt}\sum_{k=0}^{\infty}\frac{A^k t^k}{k!} = 0 + \sum_{k=1}^{\infty}\frac{kA^k t^{k-1}}{k!} = A\sum_{k=1}^{\infty}\frac{A^{k-1}t^{k-1}}{(k-1)!} = A\sum_{k=0}^{\infty}\frac{A^k t^k}{k!}$$

$$\boxed{\frac{d}{dt}e^{At} = Ae^{At}}$$

<u>Solving the controlled equation:</u> The matrix exponential plays such an important role that it has its own name: ***The State Transition Matrix***.

$$e^{A(t-t_0)} = \Phi(t,t_0)$$

$$\dot{x} = Ax \Rightarrow x(t) = \Phi(t,\tau)x(\tau)\begin{cases}\dfrac{d}{dt}\Phi(t,t_0) = A\Phi(t,t_0)\\ \Phi(t,t) = I\end{cases}$$

But if we have the controlled system: $\dot{x} = Ax + Bu$

$$x = \Phi(t,t_0)x(t_0) + \int_{t_0}^{t}\Phi(t,\tau)Bu(\tau)d\tau$$

$$x(t_0) = \underbrace{\Phi(t_0,t_0)}_{I}x(t_0) + \int_{t_0}^{t_0}\Phi(t_0,\tau)Bu(\tau)d\tau$$

$$\frac{d}{dt}x(t) = A\Phi(t,t_0)x(t_0) + \frac{d}{dt}\int_{t_0}^{t}\Phi(t,\tau)Bu(\tau)d\tau$$

$$\frac{d}{dt}\int_{t_0}^{t}f(t,\tau)d\tau = f(t,t) + \int_{t_0}^{t}\frac{d}{dt}f(t,\tau)d\tau$$

$$\Phi(t,t)Bu(t) + \int_{t_0}^{t}A\Phi(t,\tau)Bu(\tau)d\tau$$

$$\frac{d}{dt}x(t) = A\left(\Phi(t,t_0)x(t_0) + \int_{t_0}^{t}\Phi(t,\tau)Bu(\tau)d\tau\right) + Bu(t)$$

$$\frac{d}{dt}x = Ax + Bu$$

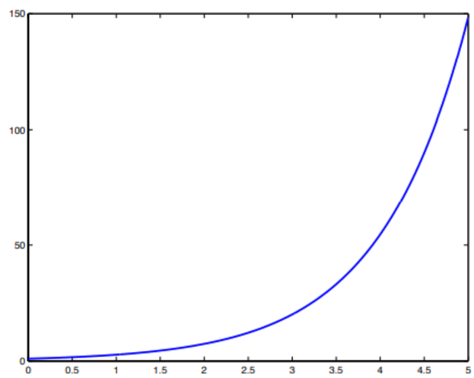
So in summary:

$$\dot{x} = Ax + Bu, y = Cx$$
$$y(t) = C\Phi(t, t_0)x(t_0) + C\int_{t_0}^{t} \Phi(t, \tau)Bu(\tau)d\tau$$
$$\Phi(t, \tau) = e^{A(t-\tau)}$$

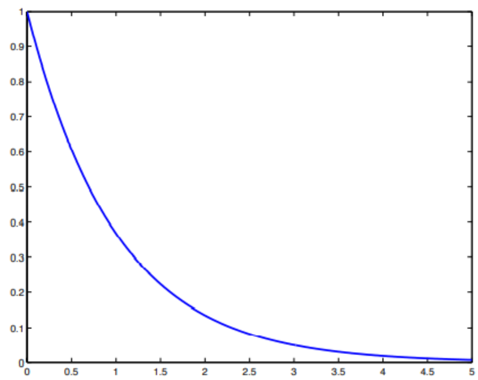
When designing a system, the first thing we try to discern is if the system is stable or not. As before, let's first consider a scalar system to understand how we can go about this.

Scalar system equation: $\dot{x} = ax \Rightarrow x(t) = e^{a(t)}x(0)$

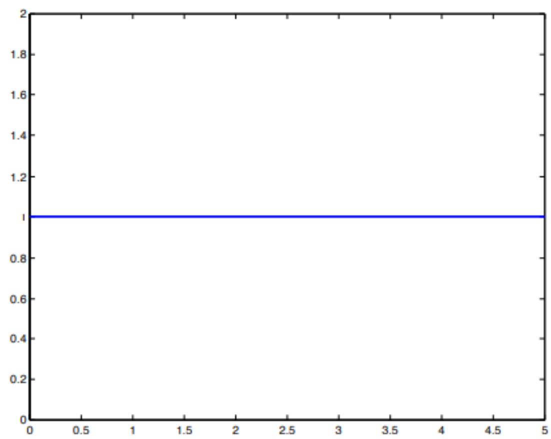
If a is assumed to be greater than 0:



If a is less than 0:



If a=0:



There are 3 cases that we need to consider in terms of stability:

- Asymptotically Stable: $x(t) \to 0, \forall x(0)$
- Unstable: $\exists x(0): \| x(t) \| \to \infty$
- Critically Stable: in-between (doesn't blow up but doesn't go to zero either)

$$\dot{x} = ax \Rightarrow x(t) = e^{a(t)}x(0)$$

$$\begin{cases} a > 0\text{: unstable} \\ a < 0\text{: asymptotically stable} \\ a = 0\text{: critically stable} \end{cases}$$

Now let's try to convert these scalars to matrices.

$\dot{x} = Ax \Rightarrow x(t) = e^{A(t)}x(0)$

We can't say for sure the A > 0 so we can try to calculate the eigen values.

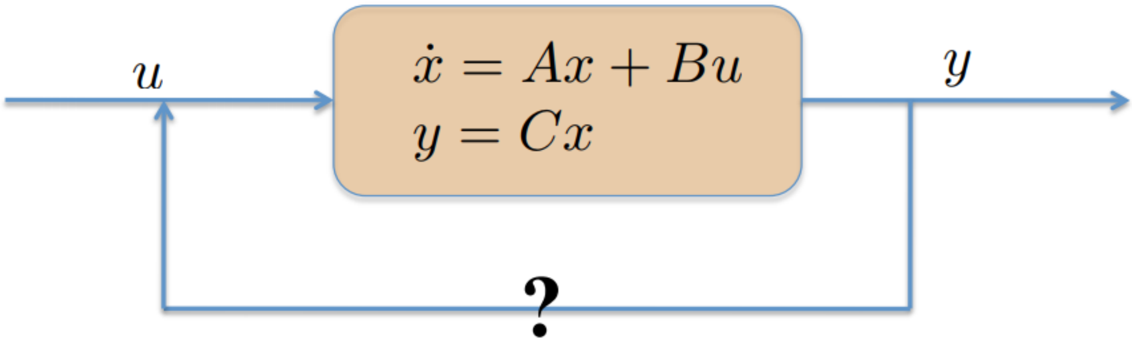
$Av = \lambda v$ where λ is the eigenvalue $\in$ C and v is the eigenvector $\in \Re^n$

These eigenvalues explain the behaviour of the matrix A in different directions (eigenvectors).

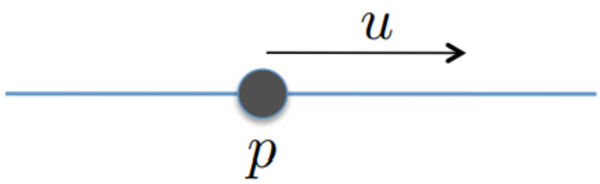
Going back to the 3 cases of stability to determine the conditions:

- Asymptotically stable (if and only if): $\text{Re}\ (\lambda) < 0, \forall \lambda \in \text{eig}\ (A)$
- Unstable (if): $\exists \lambda \in \text{eig}(A) : \text{Re}(\lambda) > 0$
- Critically stable (only if): $\text{Re}(\lambda) \leq 0, \forall \lambda \in \text{eig}(A)$
- Critically stable (if): one eigenvalue is 0 and the rest have negative real part OR two purely imaginary eigenvalues and the rest have negative real part.

We'll try to design for the asymptotically stable case. We know that all the eigenvalues have negative real part in such a scenario. So what feedback should the system have?



Let’s go back to the simplest robot to understand this.



$$u = \ddot{p}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = [1 \;\; 0] x$$

Let’s try to move towards the origin.

$u > 0$ if $y < 0$

$u < 0$ if $y > 0$ This means that: $u = -y$

In a more general sense:

$$u = -Ky = -KCx$$
$$\dot{x} = Ax + Bu = Ax - BKCx = (A - BKC)x$$

So K needs to be chosen in such a way that:

$$\mathrm{Re}\,(\lambda) < 0 \forall \lambda \in \mathrm{eig}\,(A - BKC)$$

Going back to the robot, we get:

$$\dot{x} = \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} 1 \begin{bmatrix} 1 & 0 \end{bmatrix}\right) x$$
$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x$$
$$\text{eig}\ (A - BKC) = \pm j$$

This is critically stable.

Hence, we use state feedback to stabilize the system such that all the eigenvalues have negative real part.

$$\dot{x} = Ax + Bu$$
$$u = -Kx$$
$$\dot{x} = Ax + Bu = Ax - BKx = (A - BK)x$$

Try to pick K such that the closed-loop system is stabilized

$$Re(eig(A - BK)) < 0$$

$$u \in \Re, x \in \Re^2, K: 1 \times 2$$
$$K = \begin{bmatrix} k_1 & k_2 \end{bmatrix}$$
$$\dot{x} = \left(\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix}\right) x$$
$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix} x$$

Let’s pick some values for the gains:

$$k_1 = k_2 = 1$$
$$A - BK = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$$
$$\text{eig}\ (A - BK) = -0.5 \pm 0.866j$$

This is asymptotically stable, our system will have damped oscillations.

Another set of values:

$$k_1 = 0.1, k_2 = 1$$
$$A - BK = \begin{bmatrix} 0 & 1 \\ -0.1 & -1 \end{bmatrix}$$
$$\text{eig}\,(A - BK) = -0.1127, -0.8873$$

This too is asymptotically stable but there are no oscillations.

It’s apparent now that the eigenvalues help determine the behavior of the system. Choosing the right eigenvalues is of the utmost importance.

Task:

Design a simple robot that avoids obstacles using PID controller and directly interfacing the output with the wheel velocities (differential drive model).